Sujet TD, Licence 2, module I31

Elisaberth Gavignet, Dominique Michelucci, Université de Bourgogne

Exercice 1 : Rédiger la classe PileCarac qui permet de gérer une pile de chaînes de caractères en utilisant un tableau.

L'interface de la classe est la suivante :

```
public PileCarac() ; // constructeur
public boolean pileVide(); // retourne vrai si la pile est vide et faux sinon
public boolean pilePleine() ; // retourne vrai si la pile est pleine et faux sinon
public String sommet() ; // retourne le sommet de la pile s'il existe
public PileCarac empiler(String s) ; // empile l'élément s au sommet de la pile
public PileCarac depiler() ; // retire le sommet de la pile
public String toString() ; // retourne une chaîne représentant le contenu de la pile
```

Exercice 2 : Écrire le programme qui réalise l'analyse syntaxique d'une expression arithmétique afin de déterminer si elle est correctement parenthèsée. L'expression arithmétique sera stockée dans une liste ou sous forme de chaîne de caractères et l'analyse syntaxique utilisera une pile pour stocker les parenthèses ouvertes non encore refermées. L'expression est acceptée si la pile n'est jamais vide à la lecture d'une parenthèse fermante et si la pile est vide lorsque l'expression a été entièrement parcourue.

Exercice 3 : Transformation d'une notation infixe en notation post fixe (appelée aussi notation polonaise). La notation usuelle, comme (3 + 4) * 5, est dite infixe. Son défaut est de nécessiter l'utilisation de parenthèses pour effectuer les opérations dans un ordre plutôt qu'un autre – sans les parenthèses c'est 3 + (4 * 5) qui sera effectué. Pour éviter le parenthèsage il est possible de transformer une expression infixe en expression post fixe en plaçant les opérateurs arithmétiques à la suite des opérandes auxquelles ils s'appliquent. Cette conversion est réalisable simplement en utilisant une pile dans laquelle sont stockés les opérateurs. Ces derniers ne peuvent être empilés l'un sur l'autre qui si leur priorité est croissante. Compléter le programme ci-dessous afin qu'il réalise la transformation demandée.

Exercice 4 : Proposer un programme qui permet d'évaluer le résultat d'une expression arithmétique post fixe syntaxiquement correcte. La méthode consiste à utiliser une pile pour empiler les opérandes. Ces dernières seront retirées de la pile à la rencontre d'un opérateur. Le calcul sera réalisé et le résultat empilé à son tour.

Exercice 5 : Définir une classe TriInsert qui permet de trier un ensemble de mots. Cette classe utilise trois piles (A, B et C) et une méthode triInsertion(PileCarac A, PileCarac B, PileCarac C) où PileCarac est la pile définie à l'exercice 1. Les données sont stockées dans une pile A et la méthode doit retourner une pile B contenant ces mêmes mots triés par ordre alphabétique (le plus

petit au sommet de la pile). Le principe est le suivant : on utilise la pile C qui est vide au début. Pour chacun des mots stockés dans la pile A, on considère les deux cas suivants :

– si la pile B est vide ou si l'élément au sommet de A est plus petit que celui au sommet de B : on retire l'élément au sommet de la pile A pour l'empiler dans la pile B, puis si la pile C n'est pas vide on retire tous les éléments de la pile C pour les empiler dans la pile B.

– sinon : on déplace l'élément au sommet de la pile B vers la pile C.

Exercice 6 : (pour ceux qui auront du temps) Vérifier qu'une file peut être implantée avec deux piles : une appelée « Entree », l'autre appelée « Sortie ». Les éléments insérés dans la file sont empilés dans la pile « Entree » ; les éléments qui sortent de la file sont dépilés de la pile Sortie, quand la pile Sortie n'est pas vide. Quand Sortie est vide, la pile « Entree » est vidée, par des dépilements, et chaque élément dépilé de « Entree » est empilé dans « Sortie » ; remarquez que l'ordre des éléments est inversé quand il passe de la pile « Entree » à la pile « Sortie » ; quand la pile « Sortie » n'est plus vide, il suffit de dépiler son sommet pour produire l'élément en sortie de la file. La file est vide si les deux piles sont vides. Vous assurer qu'en temps amorti, les complexités des opérations sur cette file sont bien celles attendues.